Sparcchair: A One Hundred Million Pixel Display

Bruce A. Reichlen
Sun Microsystems Laboratories, Inc.
2 Elizabeth Drive
Chelmsford, MA 01824

Abstract

We investigate whether a high-resolution head-mounted display, roving around a much larger frame buffer image, can give a user the impression of viewing a single very large display screen. We constructed a prototype consisting of an 1120 x 900 pixel head-mounted display, an ultrasonic head-tracker, a 16,384 x 6,144 pixel frame buffer, and suitable X-window control software, as a means of studying this question. Applications can write to the large frame buffer using the window system, and the viewer can navigate around the image rapidly using head rotations. To make our prototype work, changes were made to the X window display manager, such as in the positioning of user-interface objects and in the coupling between the mouse and the head tracker. User experiments were run using two applications adapted for this system and several traditional desktop applications.

The prototype system, although somewhat awkward to use due to a limited field of view in the head-mounted display, showed that head rotation is a fast, convenient way to switch display contexts, a capability that has proven to be slow and aggravating with conventional displays and window systems.

1.0 Introduction

Since the beginning of computer displays, users have wanted more information on their screens. Ivan Sutherland says "I know of no user who has access to enough pixels." This quote emphasizes the number of pixels in the display, not the display size. More pixels contain more information. Display size combined with the number of pixels determines resolution.

Our approach uses a head-mounted display for viewing a region of a much larger display area, and a head-rotation sensor for navigating the large display area. This paper will begin with a brief history of other techniques that attempt to provide users with more display information. We will then describe our approach and some experimental results. Following that, our results will be contrasted to these other techniques. Finally, we will conclude with ideas for further research.

1.1 Software Solutions

Today's window systems attempt to maximize the use of existing screen space through the use of overlapping windows and icons. This paradigm, although useful, often frustrates the user, causing him to spend an inordinate amount of time moving, resizing, and opening and closing windows. Opening an icon is frequently too slow, e.g. in the case of glancing at calendar appointments, and overlapping windows are aggravating, as in the case of cutting and pasting from one document to another.

Large virtual screens were developed to provide the user the illusion of greater screen real-estate, extending the windows model to a large virtual display area. In this model, the system creates a virtual frame buffer that is much larger than the actual frame buffer. A small map of the virtual screen is displayed at all times, providing the user with a global picture of the entire window system and a means of navigating around the virtual screen. The original paradigm intended to emulate a much larger display, where the user could freely move around this space. Although this model is a welcome extension to the windows model, the frame buffer updates take too long, and cutting and pasting between multiple documents remains slow.

A variant of virtual screens is the "rooms" paradigm, where a user's work is viewed as several different types of tasks and each task is developed or built in a different room. One room may be for writing specs, another for developing code, etc. Moving from room to room is accomplished

0-7803-1363-1/93 $3.00

through a hierarchy of doors, leading to multiple levels of rooms, and placements, which are logical representations of rooms [Henderson 86].

1.2 Hardware Solutions

One way to provide more pixels is to increase the size of the user's display. Many users do this by using multiple monitors. Several companies are developing higher resolution, larger sized monitors. These steps increase the total information content in addition to creating a larger screen. Users interact with this display using techniques similar to those in use today. One example is the Sony high-resolution 28 inch diagonal monitor. This 60 Hz non-interlaced monitor, with 1920 x 1080 pixels and a 16 x 9 aspect ratio, allows two full-size documents on the screen simultaneously [Northcutt 92]. Using this screen decreases the need for shuffling windows.

Projection technology is being used to create wall-sized displays. The "Liveboard" [Elrod 92] display is built by projecting a liquid crystal display (LCD) onto a rear projection screen. Although currently the resolution of this particular display is somewhat low, resolution will be increased as the technology matures. Even with the use of folded optics for magnifying the LCD image, the display cabinet is quite large, being approximately seven feet tall and 32 inches deep. The screen size is 46 x 32 inches. This display system is designed for group collaboration, offering more size but lower resolution.

Ultimately, technology may allow larger displays to become commonplace, but for the next several years these displays are prohibitively expensive to construct and bulky to be installed into the office environment. The Sony monitor weighs 205 pounds and is approximately 30 inches deep, so it cannot fit on a desktop. This monitor has pushed CRT technology to its limits. In the case of the Liveboard display, the unit does not fit into an office, and is not designed as a single-user system.

1.3 Our Approach

Our desire is to expand the utility of today's window systems by creating a very large display environment and an intuitive means of navigating within this environment. We wish to provide this without the expense and difficulty of a wall-sized display. Our experiments are targeted towards the individual user in an office setting who is currently using a workstation as his primary communication, information, and development platform.

Our prototype immerses a user in a cylinder, full of display information, surrounding the user 360 degrees horizontally by 135 degrees vertically. This large display area is easily navigated by tracking only the horizontal and vertical rotations of the user's head. This tracking is done at a rate of up to 30 Hz, with a lag time not exceeding 1/30 of a second. Our goal is to provide the user the illusion of an inexhaustible supply of pixels and an intuitive, natural navigation technique for viewing those pixels.

2.0 System Design

We constructed a prototype system, called "Sparcchair," which consists of a frame buffer containing approximately 100 million pixels attached to a suitably designed head-mounted display and a head-rotation sensor. This system is built into a platform containing a chair for the user. As the user moves her head, or turns in the chair, the image presented on the head-mounted display instantly changes to reflect the image corresponding to the orientation of the user's head.

Sparcchair is built by mounting a chair, modified to contain a keyboard and mouse desk, onto a platform that can swivel. The platform also holds the computer, containing a specialized frame buffer, head-mounted display, and head-rotation sensor. The X window system is modified to operate this display environment. We have used the prototype for several existing X applications and for two applications developed specifically for our prototype.

2.1 Head Mounted Display

High resolution is the primary requirement of the display needed for our prototype. Today's workstation users are accustomed to high-resolution monitors, 100 pixels per inch, and generally view

the monitor at a distance of 15 inches. This translates to an image resolution of 26 pixels per degree of viewing angle, calculated by the following formula:

100 pixels per inch / (arcsine (1 inch / 15 inches)) = 26 pixels per degree

Our display must meet this minimum standard for display resolution if we are to provide the same capabilities in today's window systems expanded to a much larger display area. The text and graphics in our windows need to be at least as crisp and clear as the systems to which our projected users are accustomed.

High display update rates are required for our system to achieve rapid navigation of the large frame buffer. Users may become uncomfortable if the update rates of the display lag the user's head movements noticeably or if image ghosting occurs when the image is altered rapidly. This constrained our choices of head mounted displays, since many of them are LCD based, which would limit the update rates and/or cause ghosting.

Because our prototype requires high resolution and fast update rates we deemed it necessary to contract a specially built display. Most of the common manufacturers in today's market, such as VPL, are concentrating on low resolution, color, LCD displays, selling into the entertainment marketplace. Although we wanted a color display, color was traded off in favor of resolution. A higher resolution "Private Eye TM" display, called the "Megapixel", specially built under contract by Reflection Technology, Inc., Waltham, MA best met our goals.

The Megapixel is an enhancement of the "Private Eye" display that has been commercially available for several years. It is a monocular display, providing an 1120 x 900 monochrome pixel image, using a 25 x 20 degree field of view, creating a very crisp, clear image in red, with a contrast ratio of approximately 35:1.This resolution is equivalent to a viewing angle of 45 pixels per degree or 168 dpi at 15 inches, easily exceeding our resolution requirements. The display is updated at 50 Hz, providing flicker-free images.

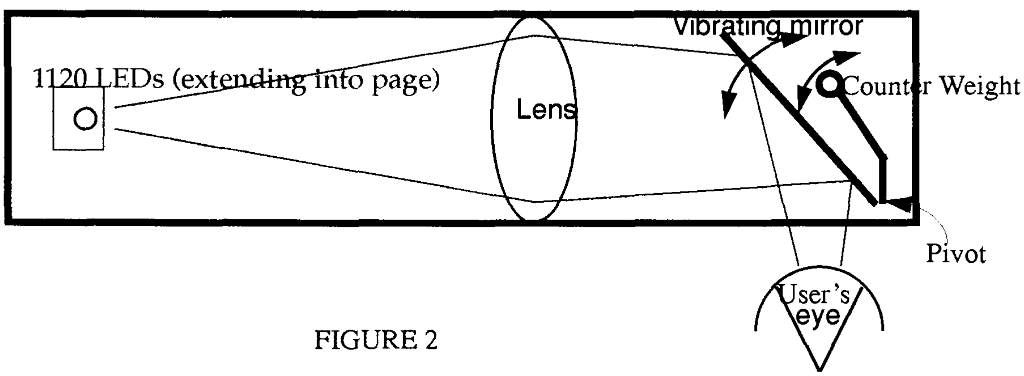

FIGURE 2

This display operates using a vibrating mirror to sweep out the image from a linear array of light emitting diodes (LEDs) as shown in Figure 2. The data on the LEDs is updated for each scan line to match the position of the mirror for sweeping the image. The mirror vibrates at 50 Hz.

2.2 Head Tracker

Our system monitors only two degrees of freedom of the user's head motion: yaw, which is rotation around the vertical axis down through the top of the head, and pitch, which is rotation around the axis through the ears. Roll, which is rotation around the axis coming out through the nose, is ignored. When seated at a desk, a user operates in a world with only two degrees of freedom. She can look or move from side to side, or tilt her head up or down. Our prototype is designed to emulate that environment.

Sparcchair incorporates Logitech's 6D head tracking system. Upon surveying the available products, we chose this one because of its update speed for measuring head rotations as well as being relatively inexpensive, easy to interface, and relatively accurate. Small gyros would have been a preferable system because of their higher update rates, but none were available for our prototype.

The Logitec head tracker is an ultrasonic device, requiring a transmitter and a receiver. Since we require 360 degrees of yaw (rotating the head side to side) and the head tracking system can track full rotation around only one axis, the receiver must be mounted on top of the user's head, and the transmitter suspended above. Although the Logitech head tracker also provides head position and roll, these measurements are ignored.

2.3 Mechanical Design

We wanted to enhance the environment of the target office user, minimizing changes as much as possible. Our environment requires the user to be seated, as at a desk. To allow the user access to the entire frame buffer, and to be comfortable while looking at data in any direction, she needs to be able to rotate easily in the chair. Our input devices are a keyboard and mouse, just as in today's window systems, so we need to keep these in front of the user at all times. Allowances are also needed to deal with cable management and to position the head tracker hardware properly.

We adapted a standard office chair for use in our experiments, as shown in Figure 1. We added a desk arm to the chair, like that of a school desk/chair, to hold the keyboard and mouse. The chair sits on a lazy susan platform, allowing the chair to swivel 270 degrees. Given the interfaces to the computer for the keyboard, mouse, head tracker, and display, it's necessary for the computer to swivel along with the chair, so the computer is placed on the platform under the chair. To the back of the platform a boom is mounted that contains the stationary transmitter of the head tracking system. The six foot boom positions the head tracker approximately 18 inches above the seated user's head.

This chair has become an essential part of the system, since without it the user would have to hold his head in a turned position to view an off center part of the frame buffer. The chair allows the user to turn easily and to keep the keyboard and mouse in front of her at all times, while keeping the user free from the entanglement of cables.

2.4 Frame Buffer Architecture

The frame buffer for Sparcchair is designed to create a display size of 360 degrees yaw x 135 degrees pitch. The number of pixels required to accomplish this is calculated as follows:

Display resolution 1120 x 900 pixels
Display field of view 25 x 22 degrees
Pixels/degree of rotation 1120 / 25 = 45 pixels / degree
Number of total degrees desired times pixels per degree
    Horizontal(Yaw) 360 * 45 = 16,200
    Vertical (Pitch) 135 * 45 = 6,075

Frame buffer width and height were rounded up to the next binary number for ease of hardware design, resulting in an actual frame buffer of 16,384 x 6,144 pixels, totaling 100.67 million pixels. The total size and resolution of the Sparcchair system is equivalent to a wall sized display larger than 8 feet by 3 feet at a viewable distance of 15 inches, providing an angular view of 45 pixels per degree.

The full 100 million pixel frame buffer is built in hardware on a 5.7 x 6.7 inch card that is plugged into a Sparcstation IPX workstation. Since the entire frame is buffered in fast memory, an arbitrary viewport can be transferred to the head-mounted display on each display refresh cycle. Although this may sound expensive, the entire frame buffer is only 12 megabytes, and uses 32 four megabit dynamic rams.

The head-mounted display interfaces directly to the frame buffer card. The region of the frame buffer sent to the display is changed simply by specifying an arbitrary upper left coordinate of the region to be viewed. The frame buffer updates the display at 50 Hz and may change the position of the displayed region for each frame.

2.5 X Window System Modifications

Modifications were made to the standard X window manager to enable it to understand the notion of "viewport." Here we use the word "viewport" to specify the region of the frame buffer currently being displayed, which is determined by where the user's head is positioned.

The standard X window manager assumes that the entire frame buffer is always displayed, but this is wrong for our approach. This assumption causes X to place dialog boxes in the middle of the frame buffer, a region that may not be in the viewport. Our new window manager, which understands viewports, places dialog boxes in the middle of the current viewport. Similarly, our new window manager causes newly created windows to appear within the viewport. Also making a window display full height required modification. We have redefined full height to be the height of the viewport instead of the frame buffer. Since our perceived pixel density is higher than on a standard 19" monitor, we also need to enlarge the default font size for readability.

Sparcchair requires a viewport manager for cursor control. When roaming around the frame buffer, the user wishes to have the cursor stay within the viewport. Current viewport position is now combined with mouse position in order to control the cursor properly, keeping it inside the viewport. When the head tracker causes the viewport to move such that the cursor would leave the viewport, the viewport manager repositions the cursor, maintaining user contact with the cursor. In effect, the mouse controls cursor position relative to the viewport.

The viewport manager also provides user controls for the head tracking system, which are continually displayed on the screen for the user to alter. A smoothing function provides stable tracking, a stick function disables tracking for small amounts of head movement, a grid function quantizes viewport motion to a user specified grid, and a freeze function enables or disables tracking.

The smoothing function reduces the effects of instability in the movement of the user's head and in the head tracker. Users are able to vary the number of rotation samples from the head tracker that are averaged when calculating viewport position. This is necessary because head movement is nonuniform and the head tracker sporadically sends bad rotation samples. Increasing the number of samples averaged reduces the effect of spurious error samples and of nonuniform head movements, but also increases the lag between actual head position and viewport position, which can interfere with coordinated movement.

Three types of sticky functions define how the tracking algorithm responds when the user begins head movement. No user holds her head completely still when gazing at a viewport, so the viewport jitters. We built some hysteresis into the system to remove the jitter. These sticky functions define how viewport movement begins when the head moves. The "Stick on Slow Motion" function establishes a threshold of head rotation velocity below which changes in rotation are ignored, leaving the viewport position unchanged. Thus if the head moves very slowly, the viewport does not move, and the viewport position no longer corresponds to absolute head rotation samples. A second alternative, "Jump on Unstick" function freezes the viewport position for small deviations in rotation angle. However once an angular threshold is crossed, the viewport will jump to the new position measured from the head sensor. This function maintains a firm coupling between the position of the viewport in the frame buffer and the position of the head. The third alternative, "Recalibrate on Unstick" is similar to Jump on Unstick, but instead of jumping to the new position calculated by the new rotation angles, it will use the angular threshold crossing as a signal that the user wishes to begin rotation. From that point, smooth position updates will be calculated from the rotations after the threshold crossing. This function maintains a loose coupling between the position of the viewport in the frame buffer and the position of the head. Stick on Slow Motion and Recalibrate on Unstick treat rotation measurements as relative, while Jump on Unstick treats them as absolute.

A grid allows the user to determine the size of the smallest viewport movement. A one-to-one relationship between the head angle and the viewing angle is maintained. The grid can be varied from one pixel to greater than a viewport size. For many applications this is more convenient than the smoothing modes.

The viewport manager displays a user-interface window that is always present on the screen. This is done by having X windows redraw the window at a new location in the frame buffer every time the viewport moves. We determined that the window must always be visible because it's essential for controlling the user environment.

In order to type text into a window, the user must click on the window to enable it. An alternative in many window systems is to enable the window that contains the cursor. Our system must use the first technique because our viewport manager frequently moves the cursor out of the current window, such as when the user looks at his keyboard. Click to type allows a window to remain active, even though the cursor moves.

The X application interface has not been altered, nor have any X internals. Existing X applications run on Sparcchair unmodified. X ported to our new frame buffer without difficulty.

3.0 Applications and Evaluations

Three different experiments were developed to test Sparcchair. Each of these is designed to test different aspects of the system: ease of navigation, viewing of time critical data, and rapid user context switching.

3.1 Map Navigation

We scanned a large, high resolution map of Boston and wrote it into the frame buffer. The user is able to navigate around the map easily and rapidly, just by turning his or her head. This experiment was created to test different head tracking controls and to evaluate rapid navigation over a large quantity of detail.

Our electronic version of this map is a great improvement over paper maps. The rapid navigation capability make following specific routes natural and intuitive. However, users have an overwhelming desire to be able to step back from the image to get a macroscopic view. Clearly a zoom out feature is required for this application.

3.2 Network Simulation

We developed a large-scale dynamic network simulation in order to experiment with navigating around a large quantity of time-critical data. The simulation focuses on data packets moving between different types of machines on a hierarchically configured network. Each machine runs pseudo applications, sending data packets of random size and at random intervals into the network. Slide bars are present at each node measuring data packets processed and queued so that bottlenecks can be spotted easily while scanning the network. An overview of the entire network is available to the user in another window usually located above the detailed network simulation.

We found the rapid and natural movement of the viewport necessary for monitoring our time-critical network simulation. The ability to scan the network quickly, observing the position of slide bars for finding bottlenecks and monitoring machine interactions is clearly a benefit.

The major disadvantage here, as in the map application, is the sense of looking through a keyhole, with a desire to step back and get a global view of the network. A small "network overview window" was added to this application as an attempted remedy. It didn't help because the overview can't show where in the network the user is looking: he has had to turn his head to look at the overview. Mechanisms for toggling between the full view and overview might help (see Section 4.1).

Finding a specific location within the network is difficult due to the restricted field of view of the display and the lack of position cues within the viewport. This difficulty appears only when the application window is significantly larger than the viewport. Most virtual reality systems have a wider field of view, reducing this difficulty, but widening the field of view without increasing the number of viewable pixels reduces resolution (pixels per degree), which makes the system poor for office applications.

3.3 General Window System Usage

Some users experimented with our system for general office work. The rapid context switching capability proves to be attractive, particularly for quick email access, or observing calendar appointments. It's a clear advantage to have an unlimited number of windows open at once, so that mouse movements and clicks are reduced. No users experienced the desire to use icons. Saving screen space is no longer an issue.

The general disadvantages experienced are the weight of the headset, the monocular screen, and the inability for the user to relax his or her eyes by momentarily gazing into the distance.

4.0 Conclusions

We have observed that users are attracted to the ease of navigation and the large number of available pixels that Sparcchair provides.

Experiments with the tracking algorithms showed that tracking yaw and pitch provides effective navigation. Most users failed to notice that Sparcchair ignored roll until we mentioned it, and even then, it was not missed. For general office work, using this display in the same manner as today's displays, two degrees of freedom are adequate, and comfortable. Most users enjoyed our system.

For certain applications, like the network simulation, total immersion in the display is appropriate and comfortable, but for other applications, such as using general office tools (electronic mail, calendar manager, etc) occluded vision even in one eye is unnatural.

Navigation within a large application proved to be difficult, due to the limited field of view of the head mounted display. Field of view is a very important navigational aid for scanning information quickly and for providing a relative sense of position within a larger space. The lack of peripheral vision also appeared to bother some users, creating a claustrophobic feeling.

Users frequently wanted to step away (zoom out) from the image. This desire seemed most prevalent when the application window is significantly larger than the viewport. We believe navigating using yaw and pitch of the user's head is so natural that they intuitively "expect" that moving away from the image should result in a zoomed out view.

This particular head mounted display has several artifacts that users found unpleasant. The eyes require time to adapt before becoming comfortable with the monocular display, and then, after a period of time (half an hour) the eye using the display grows tired. Most users complained about the vibration of the display on their heads (the vibrating mirror is quite noticeable). Some users also complained of the red color used in the display and in general no users liked the weight or bulkiness of the device. The maximum time any user continually experimented with the system was approximately two hours.

The head-rotation sensor works reasonably well in our prototype, but is occasionally unstable. During continuous smooth movement, the device sporadically sends a bad data point. This causes the wrong viewport to be displayed for 1/30 of a second. Although the effect was reduced with the addition of our smoothing function, this caused an increased lag between the viewport position and actual head position. For most users the lag did not become significantly perceptible when averaging 10 samples or fewer. Most users felt that averaging 10 samples was a good compromise between stability and lag. A filtering function needs to be added to our prototype to detect and remove samples that are unrealistic.

Generally users preferred the Jump on Unstick function over the other sticky functions. We believe this is because users wanted the absolute coupling between head direction and viewport position. Since the display we use is monocular, this absolute coupling is meaningful, allowing the user to use real world cues for positioning viewports in the displayed world.

Most users required time to grow accustomed to the required "click to type" feature.

Having the viewport manager window always present on the screen is bothersome, and because it is constantly redrawn, it takes up too many CPU cycles. A better method is needed for keeping this continually accessible but not obtrusive.

4.1 Future Work

Improvements in head-mounted display technology, yielding a more comfortable biocular device, with a wider field of view, may increase the usability of our system. Making this device partially transparent would also be beneficial, allowing the user to see his keyboard, use his telephone, or just relax his eyes.

More study is required on window systems for such a platform. The head tracking and mouse need to be much more closely tied, allowing head position to also have influence on decisions like activating, moving, or growing a window. In our prototype, the viewport manager is an application layered on top of the window manager. To be an effective solution, the viewport manager should be incorporated into the window manager, with head tracking being treated on par with mouse control.

We are proceeding with an improvement to the system that will allow rapid navigation between different viewports with the touch of a button. A set of function keys mapped to the viewport manager will vector to user-defined locations within the frame buffer. One preset function key will always vector to a screen containing a world view map, analogous to the world view in virtual open windows, and the viewport manager. A second toggle of this key will return the user to the previous viewport. We are hopeful that this will make overview maps more useful, since now the currently active viewport can be displayed and even moved on the overview.

5.0 Acknowledgments

Several people contributed ideas to this effort. It was Ivan Sutherland's idea to build a large display using a 100 million pixel frame buffer with a head-mounted display and head-rotation tracking. Bruce Reichlen contributed the system, hardware, and rotating chair designs. The tracking functions, X window modifications and application development were done by Ralph Marshall and Peter Jones. Tony Staiti provided mechanical and electrical construction. Without Bob Sproull's support and encouragement, this paper would not have made it to print.

6.0 References

Card, S.K., and Henderson, D.A., Jr. [1987] *A Multiple, Virtual Workspace Interface to Support User Task Switching*, CHI+GI 1987 Conference Proceedings on Human Factors in Computing Systems and Graphics Interface, (April, 1987) pp. 53-39.

Henderson, D.A., Jr., and Card, S.K., [1986] *Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface*, ACM Transactions on Computer Graphics, Vol 5, No. 3 (July 1986) pp. 210-243.

Lin, J. K., [1992] *Virtual Screen: A Framework for Task Management*, X Technical Conference Proceedings (January 1992) pp 191-197.

Becker, A., [1990] *Miniature Video Display System*, US Patent number 4,934,773 (June 1990).

Northcutt, J.D., Wall, G. A., Hanko, J. G., and Kuerner, E. M., [1992] *A High Resolution Video Workstation*, Sun Microsystems Laboratories, Inc., SMLI 92-0056.

Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., and Welch, B., [1992] *Liveboard: A large interactive Display Supporting Group Meetings, Presentations and Remote Collaboration*, CHI+GI 1992 Conference Proceedings on Human Factors in Computing Systems and Graphics Interface, (May, 1992) pp. 599-607.